TASK3- CLASSIFICATION PROBLEM

OBJECTIVE-: Use the data available for the first three days, since the user signed up for the 7 day trial, to determine the target variable i.e. the paid_conversion_score (How likely the user is to convert to a paid subscription after their trial is complete)

Predict if a client will subscribe (yes/no) from trial subscription to a paid subscription — this is defined as a classification problem and evaluate the paid_conversion_score(likability of user to become paid subscriber).

DATA SET DESCRIPTION- The given data contains 4 tables containing the information of the subscribed user data. The information given is regarding the subscription details of the user for the trial subscription , the content watched over the trial subscription period and other pointers related to its related details, Profile based usage can also be found and the last table describes the birth year and zipcode of the signed up user. The features that we have represents the Sign-up user behavior, statistics, demographic analysis, product data, payment details, etc to help indicate and information retrieved can be used to make various business and policy decisions for targeting specific users, geographic location etc., to help increase the market.

1. The tables given would be associated with the number of records/instances given inorder to estimate the size of the data given(No. of rows)
2. It contains a mix of what kind of attributes- numerical, categorical, date , time etc., has to be recorded.

DATA PREPARATION AND LOADING- We can aggregate all the 4 data tables given for the first three days when the user signed up for 7 day trial subscription using the "registration_id" of the signed up user and load it in appropriate data structure like pandas data frame in order to do the further analysis for the exploration.

DATA EXPLORATION- This is the very important phase of understanding the given data and the features and other important latent/hidden knowledge that we can draw from the given features to understand more useful information for the solving the problem.

This step also enables the business to retrieve some very fascinating details regarding the consumer behaviour and demographic analysis and will also help understand the factors behind most paid subcriptions. Other very important use is to use the retrieved information to make useful policy decisions for future business habits geographically and based on specific consumer habits and analysis. Like identify top countries, top products and contents users were interested most in, what kind of user, age range of most paid subscribers etc.

ACTUAL TARGET(Given)-converted_ind- Is the True or actual indicator if that particular user or registration_id gets converted from trial to paid subscription.

PREDICTED TARGET(To be predicted while classification)- Lets create a new target variable "predicted_converted_ind" for binary classification that we will use for storing the results of the predictions if the specific user will be converted into the paid subscriber(TRUE) or not(FALSE).

1. Type inference: detect the types of columns in a dataframe.
2. Missing values in each column

3. Unique values
4. Quantile statistics like minimum value, Q1, median, Q3, maximum, range, interquartile range
5. Descriptive statistics like mean, mode, standard deviation, sum, median absolute deviation, coefficient of variation, kurtosis, skewness
6. Most frequent values
7. Histograms
8. Correlations highlighting of highly correlated variables, Spearman, Pearson and Kendall matrices
9. Duplicate samples/rows
10. Identifying the NAs , NAN etc.
11. How many features are numerical and how many are categorical etc.
12. Visualize the graphical representation of the distribution of age of the trial subscribed user to paid subscribed user as well to find out relevant age groups which are more likely to become a paid subscriber(like family, individual, youth(student) , old, etc) (we can plot box plot, frequency bar plot, distribution
13. Filter data which passes certain wrong rules.

We can make very interesting and powerful visualization to present these under-represented facts/insights from the data. Like some that I would be interested in drawing are as below:

DATA EXPLORATION:

Evaluate the Prevalence of positive class (paid subscribers). Percentage of subscribers who resulted in the actual paid subscription in the given data. (to check the imbalance in the data as well for the classification purposes) . Various data visuals and evaluations of facts and figures might be useful for the business.

1.How many users who became subscribed users ,used a specifc device- were adults or kids

2. which signup place has been most converted to subscribed users

3. does campaigning bring more subscribed users

4. How many users used coupons to get converted to subscribed

5.what kind of unique content or video id fetched most subscribers

6.correlation of video_content_complete with true subscribed users

7.how many subscribed users were due to kids and adults profiles

8.which location/zipcode or country has most subcribed users and find out if they are adult or kids

9. age of most signup user to subscibed user

10. Geolocation visualization to indicate the actual paid subscribers and subscribers who cancel before the paid subscription like makers indicating visual location on map geo-location , state, country , city.

11. Which signup plan(signup_plan_cd). has turned out subscribers to be more to become a paid subscriber.

12. Which coupon code brought most subscriptions(highly popular) and which coupon codes turned out to generate paid subscribers

13. Which plan is signed up via using which kind of device so that we can incorporate this information to recommend more of that kind of plans to future subscribers. Similarly coupons.

14. Which kind of campaigns brought most paid signups, what devices were used for sign ups. Also try to measure the user response times in the signups is fastest from which kind of campaigns

15. Find out which kind of campaigns brought most win backs and most new users.

16. What is the ratio of winbacks vs new users to become a paid subscriber.

17. What kind of shows , genre, seasons, video category etc brought more signups and most paid subscribers.

18. What kind of videos have been watched by what age groups and which geo-locations, by what kind of profiles(adults, kids)

19. Which country, state , city has highest number of subscribers(paid) obtained after mapping zipcode to subsequent country, state and cities.

20. We can create a pie chart to show the percentage use of each kind of devices overall.

21. We can also produce pie chart of the amount of time spent on the content on day to day basis to understand the amount of user engagement with the content. Similarly we can also plot the use of content watching device overtime in percentages.

22. Graphical representation of categorical features

23. Outlier analysis

FEATURE ENGINEERING-

1. Convert or map zipcode to country, state, city for better visualization or assessment of paid subscribers .
2. Do feature selection techniques to find out most useful features to use and to remove the highly collinear features to remove multicollinearity from the data.
3. Generate some more useful features like-
   i) Difference in actual subscription price – Price after applying discount code = price paid by subscribers to become highly likely paid subscriber. OR likability of discount code to become a more likely subscriber.
   ii) Find out percentage of users who have completely watched the content
   iii) Find out percentage users who have watched 25% , 75% of the content
   iv) Number of unique videos watched by a profile and registration ids evaluated using the video_id
   v) Evaluate the distribution of Total number of content watched by a specific user using the kde(kernel density estimation plot)
   vi) Evaluate the distribution of total number of content watched till the completion
   vii) May be we can convert date of sign up to more useful feature of “is_weekend” or not.

DATA PRE-PROCESSING-

1. Remove or drop un-necessary columns/features that is not required as per modelling development for the classification process:
   trial_start_dt, registration_id, video_id
2. Drop highly correlated features so as to make sure we don’t have multi-collinearity in our data.
3. Perform the one-hot encoding of the categorical features

4. Perform Scaling the attributes the numerical features to be scaled using standard scaler.

MODEL DEVELOPMENT-

Building Training, Validation & Test Samples
So far we have explored our data and created features from the categorical data. It is now time for us to split our data. The idea behind splitting the data is so that you can measure how well your model would do on unseen data. We split into three parts:

Training samples: these are samples from the data set used to train the model. It can be 70% of the data.

Validation samples: these are samples used to validate or make decisions from the model. It can be 15% of the data.

Test samples: these are samples used to measure the accuracy or performace of the model. It can be 15% of the data.

In this project, we will split into 70% train, 15% validation, and 15% test.

Let's shuffle the samples using sample in case there was some order (e.g. all positive samples on top). Here n is the number of samples. random_state is just specified so the project is reproducable.

1. Test the prevalence in each of the training , validation and test sets.
2. Balance the dataset between the positive and negative classes- We would need to balance the number of samples of positive and negative classes as are the the target output labels for this specific task is. In order for our algorithm to be less biased with the prediction of more prevalent class.
   Two approaches – ***Undersampling and Oversampling*** are techniques which can be employed to suffix the task.
3. Balancing will keep the prevalence of all three train, validation and test sets to 0.5 or 50%.
4. Impute missing values in the data if any.
5. Apply scaling mechanisms if applicable by respective algorithms we choose to fit our data on.

This section allows us to test various machine learning algorithm to see how our independent variables accurately predict our dependent output variable. We will then select the best model based on performance on the validation set.

Model Selection:

In this section, we will first compare the performance of the following machine learning models using default hyperparameters:

-LOGISTIC REGRESSION: Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Logsitic regression uses a line (Sigmoid function) in the form of an "S" to predict if the dependent variable is true or false based on the independent variables. The "S-shaped" curve (on the line graph) will show the probability of the dependent variable occuring based on where the points of the

independent variables lands on the curve. In this case, the output is predicted by the numerical and categorical variables.

-STOCHASTIC GRADIENT DESCENT- Stochastic Gradient Descent analyzes various sections of the data instead of the data as a whole and predicts the output using the independent variables. Stochastic Gradient Descent is faster than logistic regression in the sense that it doesn't run the whole dataset but instead looks at different parts of the dataset.

-BAGGGED DECISION TREE:

Bagging (Bootstrap Aggregation) is used when our goal is to reduce the variance of a decision tree. Here idea is to create several subsets of data from training sample chosen randomly with replacement. Now, each collection of subset data is used to train their decision trees.

Pros- Bagging is used with decision trees, where it significantly increases the stability of models in the reduction of variance and improving accuracy, which eliminates the challenge of overfitting. Bagging in ensemble machine learning takes several weak models, aggregating the predictions to select the best prediction.

-RANDOM FOREST- It is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

Advantages-

Random forests is great with high dimensional data since we are working with subsets of data. It is faster to train than decision trees because we are working only on a subset of features in this model, so we can easily work with hundreds of features.

-GRADIENT BOOSTING CLASSIFIER- Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision tree. GBT build trees one at a time, where each new tree helps to correct errors made by previously trained tree.

Main Features:

It performs the optimization in function space (rather than in parameter space) which makes the use of custom loss functions much easier.

Boosting focuses step by step on difficult examples that give a nice strategy to deal with unbalanced datasets by strengthening the impact of the positive class(if applicable).

Strengths of the model-

Since boosted trees are derived by optimizing an objective function, basically GBM can be used to solve almost all objective function that we can write gradient out.

Weaknesses of the model-

GBMs are more sensitive to overfitting if the data is noisy.

Training generally takes longer because of the fact that trees are built sequentially.

GBMs are harder to tune than RF. There are typically three parameters: number of trees, depth of trees and learning rate, and each tree built is generally shallow.

- **XG BOOST CLASSIFIER:** XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

Three main forms of gradient boosting are supported:

Gradient Boosting algorithm(or GBM),Stochastic Gradient Boosting with sub-sampling at the row, column and column per split levels, *Regularized Gradient Boosting* with both L1 and L2 regularization.

Benefits over Gradient boosting-

1. Execution Speed.
2. Model Performance.

During the model development process we will use the following -

1. Using the k-fold cross-validation and measure the cross-validation score,accuracy, precision, recall, f1 , AUC
2. Perform Hyper-parameter tuning for better optimization using Grid-search cv and evaluate the best parameters and the corresponding measures on the training and the validation data sets.
3. Choose the model with the best performance. Apart from evaluating the performance via Accuracy we will also give high weightage to AUC score for this problem as it would be a good score to evaluate the performance of the target variable in this case than the true positive rate and false positive rate.
4. We can evaluate the learning curves for the models on the validation sets
5. Find the feature importances .
6. Predict the probability of the output predicted labels(True o False for each user to become a likely paid subscriber) and other probabilities (if required)

MODEL EVALUATION-

The following measures we will use to evaluate the performance of the best selected model on the training ,validation and test sets and will select the best model:

1. Accuracy
2. Recall
3. Precision
4. AUC(Area under the curve)- In this project, we will utilize the Area under the ROC curve (AUC) to evaluate the best model. This is a good data science performance metric for picking the best model since it captures the trade off between the true positive and false positive and does not require selecting a threshold.
   AUC (area under the ROC curve) as a performance indicator. The reason I chose this over other indicators such as precision and accuracy is that it measures the relationshio between true positives

and false positives in our data in order to derive a score that depicts that. Also, AUC is widely used and an easier metric to compare many models with.

5. F1
6. Prevalence is kept at 0.5 for making both the positive and negative cases equally probable by balancing the imbalance in the dataset.
7. Specificity

-In order to evaluate the likelihood of the trial subscriber to become a paid subscriber by the end of the trial (paid_conversion_score) can be evaluated by evaluating the final predicted target label's probability values in applicable cases in order to estimate the likelihood of the prediction made.

-Find the feature importances . We can also find features with positive correlation with the output and negative features which contributed negatively in the classification process.

CONCLUSION-

We will use the AUC score in combination of some other measures than to check and access the overall performance of the task. Recall will help us to know how much of the true positives were identified or captured correctly by our machine learning model.

Paid_conversion_score is also evaluated during the process to predicting the likelihood of the target classification made by the model.

While modelling we need to see if there is no overfitting and also find out if there is variance and bias present in the model then take appropriate steps to correct the same or choose the right decision in terms of modelling otherwise.

There are various kinds of information which can be used to leverage the demographic, behavioral and engagement related facts and can also provide some very useful insights for the business to make various policy and other decisions.

****

*Note – The time taken for TASK 1 and TASK 2 is ~30 min . The average Running Times on my computer with i7 processor and 12 gb ram is 0.036 sec for task1 and 0.21 seconds for task2 respectively.